

ESTRUCTURAS DE CONTROL

EN LENGUAJES DE PROGRAMACIÓN, LAS ESTRUCTURAS DE CONTROL PERMITEN MODIFICAR EL FLUJO DE EJECUCIÓN DE LAS INSTRUCCIONES DE UN PROGRAMA.

CON LAS ESTRUCTURAS DE CONTROL SE PUEDE:

- DE ACUERDO A UNA CONDICIÓN, EJECUTAR UN GRUPO U OTRO DE SENTENCIAS (IF-THEN-ELSE Y SELECT-CASE)
- EJECUTAR UN GRUPO DE SENTENCIAS MIENTRAS EXISTA UNA CONDICIÓN (DO-WHILE)
- EJECUTAR UN GRUPO DE SENTENCIAS HASTA QUE EXISTA UNA CONDICIÓN (DO-UNTIL)
- EJECUTAR UN GRUPO DE SENTENCIAS UN NÚMERO DETERMINADO DE VECES (FOR-NEXT)
- ETC.

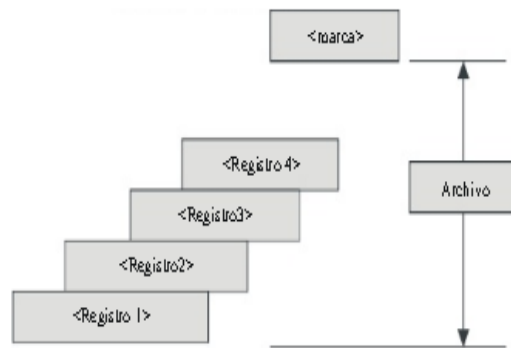
ESQUEMA CUANTITATIVO

EL ESQUEMA CUANTITATIVO ES UTILIZADO CUANDO SE CONOCE EL NÚMERO DE VECES QUE DEBE REPETIRSE UN CICLO DETERMINADO, ANTES DE ACTIVARSE LA ESTRUCTURA REPETITIVA.



ESQUEMA CUALITATIVO

ESTE ESQUEMA ES UTILIZADO CUANDO NO SE CONOCE EL NÚMERO DE VECES QUE DEBE REPETIRSE LA SECUENCIA DE UN DETERMINADO CICLO. EN ESTA CLASE DE CICLO EL USUARIO DECIDE CUÁNDO TERMINAR CON EL PROCESO REPETITIVO SIN IMPORTAR CUANTAS ITERACIONES SE HAYAN EFECTUADO; A DIFERENCIA DE LOS CICLOS ANTERIORES, EN DONDE EL NÚMERO DE ITERACIONES ESTÁ DETERMINADO DESDE EL PRINCIPIO.



ESTRUCTURA MIENTRAS

LA ESTRUCTURA REPETITIVA MIENTRAS ES AQUELLA EN QUE EL CUERPO DEL BUCLE SE REPITE MIENTRAS SE CUMPLE UNA DETERMINADA CONDICIÓN. CUANDO SE EJECUTA ESTA INSTRUCCIÓN, LA PRIMERA COSA QUE SUCEDE ES QUE SE EVALÚA LA CONDICIÓN. SI LA EXPRESIÓN ES VERDADERA, ENTONCES SE EJECUTA EL CUERPO DEL BUCLE. ESTE PROCESO SE REPITE UNA Y OTRA VEZ MIENTRAS LA CONDICIÓN SEA VERDADERA.

Repita el cuerpo del bucle mientras se cumpla una determinada condición.

Pseudocódigo	Sintaxis en C++	Programa en C++
Mientras {condición} Acción_1; Acción_2; Acción_n; Fin_mientras Instrucción X	while (exp_prueba) instrucción; while (exp_prueba) { Instrucción_a; Instrucción_b; Instrucción_n; }	<pre>#include <iostream.h> #include <stdlib.h> int main() { int inumtab, i=1; cout << "\n Numero de la Tabla de Multiplicar: "; cin >> inumtab; while (i<=10){ cout<< inumtab << " * " << i << " = "; cout<< inumtab<<endl; i++; } system("PAUSE"); return 0; }</pre>

EJEMPLO

Si: **for** (expresión 1; expresión 2; expresión 3) sentencia

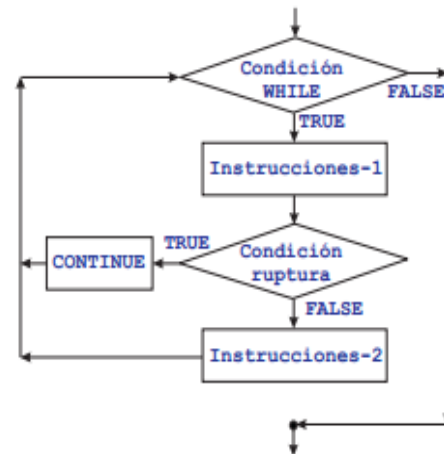
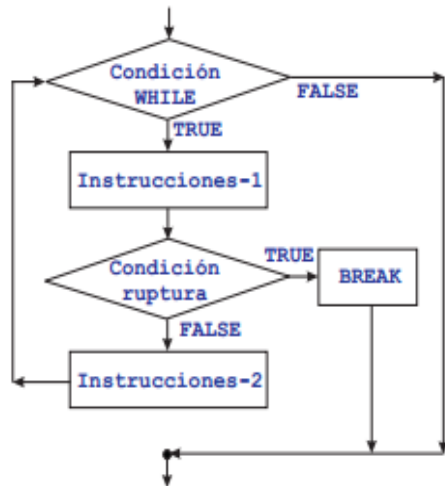
Con el **while** sería:

```
Expresión 1;  
while (expresión 2) {  
    Sentencia  
    expresión 3;  
}
```

RUPTURA DE CICLOS

DENTRO DE LA PROGRAMACIÓN, ALGUNAS VECES, ES NECESARIO HACER QUE UN CICLO SE DETENGA ABRUPTAMENTE (ABORTE), PORQUE YÁ SE CUMPLIÓ ALGO QUE SE NECESITABA O SE ESTABA BUSCANDO, POR LO QUE, POSIBLEMENTE, NO SE ALCANCE A SATISFACER COMPLETAMENTE EN UNA FORMA NORMAL LA CULMINACIÓN DE LA CANTIDAD DE VECES QUE DEBE EJECUTARSE O REPETIRSE UN CICLO. LA RUPTURA SE HACE CAMBIANDO EL SENTIDO DE LA EXPRESIÓN LÓGICA QUE CONTROLA EL CICLO, PARA QUE ÉSTA SEA FALSA Y NO SE

CONTINÚE EJECUTANDO LA SECUENCIA DE INSTRUCCIONES.



ESTRUCTURA PARA

EN C/C++ SE FORMA UNA ESTRUCTURA UTILIZANDO LA PALABRA RESERVADA STRUCT, SEGUIDA POR UN CAMPO ETIQUETA OPCIONAL, Y LUEGO UNA LISTA DE MIEMBROS DENTRO DE LA ESTRUCTURA. LA ETIQUETA OPCIONAL SE UTILIZA PARA CREAR OTRAS VARIABLES DEL TIPO PARTICULAR DE LA ESTRUCTURA:

INICIO

PARA $F \leftarrow 1$ HASTA $F \leq 100$ A
INCREMENTOS DE $F \leftarrow F + 1$ HACER
ESCRIBIR (F)

FIN _ PARA

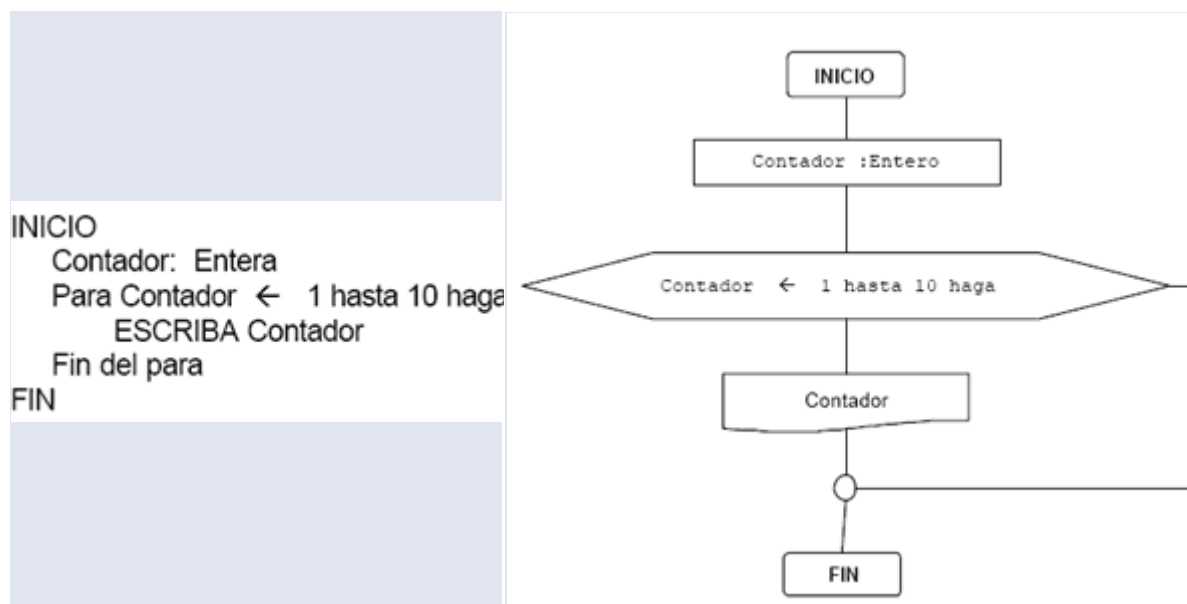
FIN

EJEMPLO

REALIZAR UN ALGORITMO QUE MUESTRE LOS NÚMEROS DE UNO EN UNO HASTA DIEZ USANDO UNA ESTRUCTURA PARA. EXPRESE EL ALGORITMO USANDO PSEUDOCÓDIGO Y DIAGRAMA DE FLUJOS.

Pseudocódigo

Diagrama de Flujos



ESTRUCTURA MIENTRAS QUE

LA ESTRUCTURA “HACER-MIENTRAS” ES UNA SENTENCIA DEL TIPO

“REPITA MIENTRAS” CON LA DIFERENCIA QUE EVALÚA LA

CONDICIÓN AL FINAL DEL LAZO Y NO AL PRINCIPIO.

LAS SENTENCIAS (UNA O MÁS) DEL CUERPO DEL LAZO SE EJECUTAN

MIENTRAS QUE LA CONDICIÓN (EXPRESIÓN LÓGICA) ES CIERTA.

Código en C
<pre>do { printf("A es mayor \n"); } while (A>=B)</pre>
<pre>int contador = 1 int exponente = 0 do { contador *= 2; exponente++; } while (contador <= 2048) if (pow(2,exponente) == contador) printf("resultado correcto");</pre>

EJEMPLO

Representación gráfica:

